IN THE CLAIMS:

Please amend the claims such that the pending claims read as follows:

1. (Cancelled)

2. (Cancelled)

SUB B77

3. (New) A method for recording a plurality of data about a plurality of blocks of data stored in a storage system, comprising the steps of:

maintaining multiple usage bits for each of said plurality of blocks, wherein one bit of said multiple usage bits for each of said plurality of blocks indicates a block's membership in an active file system and one or more bits of said multiple usage bits for each of said plurality of blocks indicate membership in one or more read-only copies of a file system; and

A2

storing, in said storage system, said multiple usage bits for each of said plurality of blocks.

4. (New) A method as in claim 3, wherein one or more bits of said multiple usage bits for each of said plurality of blocks further indicate block reusability.

a

5. (New) A method for generating a consistency point for a storage system, comprising the steps of:

marking a plurality of inodes pointing to a plurality of modified blocks in a file system stored on said storage system as being in a consistency point;

flushing regular files to said storage system;

flushing special files to said storage system;

flushing at least one block of file system information to said storage system; and

requeueing any dirty inodes that were not part of said consistency point.

6. (New) A method as in claim 5, wherein said step of flushing said special files to said storage system further comprises the steps of:

A2

pre-flushing an inode for a blockmap file to an inode file;

allocating space on said storage system for all dirty blocks in said inode and said blockmap files;

flushing said inode for said blockmap file again;

updating a plurality of entries in said blockmap file wherein each entry of said plurality of entries represents a block in said storage system; and

writing all dirty blocks in said blockmap file and said inode file to said storage system.

a

7. (New) A method of maintaining data in a storage system, comprising the steps of:

maintaining a root node and inodes for a file system, the root node pointing directly or indirectly to the inodes, and each inode storing file data, pointing to one or more blocks in the storage system that store file data, or pointing to other inodes;

maintaining an inode map and a block map for the file system; and

after data in the file system is changed, writing new data to one or more new blocks in the storage system, maintaining old data in old blocks in the storage system, updating the inodes and inode map to reflect the new blocks, and updating the block map, with the block map showing that both the new blocks and the old blocks are in use;

whereby a record of changes to the file system is automatically maintained in the storage system.

8. (New) A method as in claim 7, further comprising the step of temporarily storing the new data and inodes affected by the new data in memory before writing the new data and inodes affected by the new data to the storage system.

9. (New) A method as in claim 8, further comprising the step of using a list of dirty inodes to coordinate writing the new data and inodes affected by the new data to the storage system.

SUB B87

10. (New) A method as in claim 7, further comprising the step of creating a snapshot of the file system by copying the root node.

11. (New) A method as in claim 10, wherein the block map indicates membership of blocks in one or more snapshots.

12. (New) A method as in claim 10, further comprising the step of deleting a snapshot from the storage system, wherein blocks that are only part of the deleted snapshot are released for re-use by the storage system.

A2

13. (New) A memory storing information including instructions, the instructions executable by a processor to record a plurality of data about a plurality of blocks of data stored in a storage system, the instructions comprising the steps of:

maintaining multiple usage bits for each of said plurality of blocks, wherein one bit of said multiple usage bits for each of said plurality of blocks indicates a block's membership in an active file system and one or more bits of said multiple usage bits for each of said plurality of blocks indicate membership in one or more read-only copies of a file system; and

storing, in said storage system, said multiple usage bits for each of said plurality of blocks.

14. (New) A memory as in claim 13, wherein one or more bits of said multiple usage bits for each of said plurality of blocks further indicate block reusability.

15. (New) A memory storing information including instructions, the instructions executable by a processor to generate a consistency point for a storage system, the instructions comprising the steps of:

marking a plurality of inodes pointing to a plurality of modified blocks in a file system stored on said storage system as being in a consistency point;

flushing regular files to said storage system;

flushing special files to said storage system;

flushing at least one block of file system information to said storage system; and

requeueing any dirty inodes that were not part of said consistency point.

16. (New) A memory as in claim 15, wherein said step of flushing said special files to said storage system further comprises the steps of:

pre-flushing an inode for a blockmap file to an inode file;

allocating space on said storage system for all dirty blocks in said inode and said blockmap files;

flushing said inode for said blockmap file again;

updating a plurality of entries in said blockmap file wherein each entry of said plurality of entries represents a block in said storage system; and

writing all dirty blocks in said blockmap file and said inode file to said storage system.

17. (New) A memory storing information including instructions, the instructions executable by a processor to maintain data in a storage system, the instructions comprising the steps of:

maintaining a root node and inodes for a file system, the root node pointing directly or indirectly to the inodes, and each inode storing file data, pointing to one or more blocks in the storage system that store file data, or pointing to other inodes;

maintaining an inode map and a block map for the file system; and

after data in the file system is changed, writing new data to one or more new blocks in the storage system, maintaining old data in old blocks in the storage system, updating the inodes and inode map to reflect the new blocks, and updating the block map, with the block map showing that both the new blocks and the old blocks are in use;

whereby a record of changes to the file system is automatically maintained in the storage system.

18. (New) A memory as in claim 17, wherein the instructions further comprise the step of temporarily storing the new data and inodes affected by the new data in memory before writing the new data and inodes affected by the new data to the storage system.

19. (New) A memory as in claim 18, wherein the instructions further comprise the step of using a list of dirty inodes to coordinate writing the new data and inodes affected by the new data to the storage system.

SUB B97

20. (New) A memory as in claim 17, wherein the instructions further comprise the step of creating a snapshot of the file system by copying the root node.

21. (New) A memory as in claim 20, wherein the block map indicates membership of blocks in one or more snapshots.

A2

22. (New) A memory as in claim 20, wherein the instructions further comprise the step of deleting a snapshot from the storage system, wherein blocks that are only part of the deleted snapshot are released for re-use by the storage system.

23. (New) A system comprising:

a processor;

a storage system; and

a memory storing information including instructions, the instructions executable by the processor to record a plurality of data about a plurality of blocks of data stored in the storage system, the instructions comprising the steps of: (a) maintaining multiple usage bits for each of said plurality of blocks, wherein one bit of said multiple usage bits for each of said

plurality of blocks indicates a block's membership in an active file system and one or more bits of said multiple usage bits for each of said plurality of blocks indicate membership in one or more read-only copies of a file system; and (b) storing, in said storage system, said multiple usage bits for each of said plurality of blocks.

24. (New) A system as in claim 13, wherein one or more bits of said multiple usage bits for each of said plurality of blocks further indicate block reusability.

25. (New) A system comprising:

a processor;

a storage system; and

a memory storing information including instructions, the instructions executable by the processor to generate a consistency point for the storage system, the instructions comprising the steps of: (a) marking a plurality of inodes pointing to a plurality of modified blocks in a file system stored on said storage system as being in a consistency point; (b) flushing regular files to said storage system; (c) flushing special files to said storage system; (d) flushing at least one block of file system information to said storage system; and (e) requeueing any dirty inodes that were not part of said consistency point.

26. (New) A system as in claim 25, wherein said step of flushing said special files to said storage system further comprises the steps of: (f) pre-flushing an inode for a blockmap file

to an inode file; (g) allocating space on said storage system for all dirty blocks in said inode and said blockmap files; (h) flushing said inode for said blockmap file again; (i) updating a plurality of entries in said blockmap file wherein each entry of said plurality of entries represents a block in said storage system; and (j) writing all dirty blocks in said blockmap file and said inode file to said storage system.

27. (New) A system comprising:

a processor;

a storage system; and

a memory storing information including instructions, the instructions executable by the processor to maintain data in the storage system, the instructions comprising the steps of: (a) maintaining a root node and inodes for a file system, the root node pointing directly or indirectly to the inodes, and each inode storing file data, pointing to one or more blocks in the storage system that store file data, or pointing to other inodes; (b) maintaining an inode map and a block map for the file system; and (c) after data in the file system is changed, writing new data to one or more new blocks in the storage system, maintaining old data in old blocks in the storage system, updating the inodes and inode map to reflect the new blocks, and updating the block map, with the block map showing that both the new blocks and the old blocks are in use;

whereby a record of changes to the file system is automatically maintained in the storage system.

28. (New) A system as in claim 17, wherein the instructions further comprise the step of temporarily storing the new data and inodes affected by the new data in memory before writing the new data and inodes affected by the new data to the storage system.

29. (New) A system as in claim 18, wherein the instructions further comprise the step of using a list of dirty inodes to coordinate writing the new data and inodes affected by the new data to the storage system.

SUB B10 30. (New) A system as in claim 17, wherein the instructions further comprise the step of creating a snapshot of the file system by copying the root node.

A2 31. (New) A system as in claim 20, wherein the block map indicates membership of blocks in one or more snapshots.

32. (New) A system as in claim 20, wherein the instructions further comprise the step of deleting a snapshot from the storage system, wherein blocks that are only part of the deleted snapshot are released for re-use by the storage system.

33. (New) A system for maintaining data in storage means, comprising the steps of:

means for maintaining a root node and inodes for a file system, the root node pointing to inodes, and each inode storing file data, pointing to one or more blocks in the storage means that store file data, or pointing to other inodes;

means for maintaining an inode map and a block map for the file system; and

means for, after data in the file system is changed, writing new data to one or more new blocks in the storage means, maintaining old data in old blocks in the storage means, updating the inodes and inode map to reflect the new blocks, and updating the block map, with the block map showing that both the new blocks and the old blocks are in use;

whereby a record of changes to the file system is automatically maintained in the storage means.